

IN THE CLAIMS:

Please cancel claims 13, 14 and 31, and amend the claims as follows:

1. (Currently Amended) A computer-implemented method [[for]] of execution of a multi-step workflow that is repeatedly executed on data of a database, wherein the workflow is defined by a plurality of steps, each step being an executable function that operates on input from a previous step and produces output for a subsequent step and each step having a defined input format and output format, the method comprising:

receiving current input for execution of a step of the workflow on relevant data of the database, wherein the step has been previously executed on the relevant data using previous input identical to the current input and wherein the previous execution of the step produced previous output;

determining whether the step is deterministic, whereby the step generates identical output for given input in repeated executions of the step on the relevant data; and

if the step is deterministic, returning the previous output produced during the previous execution of the step without re-executing the step.

2. (Original) The method of claim 1, further comprising using the returned previous output as input to a next sequential step in the workflow.

3. (Original) The method of claim 1, wherein determining whether the step is deterministic comprises determining whether a workflow description of the step includes a deterministic flag indicating that the step generates identical output for given input in repeated executions of the step on the relevant data.

4. (Original) The method of claim 1, further comprising:
determining whether the current input and the previous input are the same; and

returning the previous output produced during the previous execution of the step only if the current input and the previous input are determined to be the same.

5. (Original) The method of claim 4, wherein determining whether the current input and the previous input are the same comprises accessing a hash table representative of the previous input.

6. (Original) The method of claim 1, further comprising:
determining whether the relevant data has been changed since the previous execution; and
returning the previous output produced during the previous execution of the step only if the relevant data has not been changed.

7. (Original) The method of claim 6, wherein determining whether the relevant data has been changed comprises:
determining a timestamp indicating a point of time of the previous execution; and
determining, from a transaction log of the database, whether transactions relative to the relevant data have occurred since the point of time indicated by the timestamp.

8. (Currently Amended) The method of claim 6, further comprising:
if the relevant data has been changed since the previous execution:
executing the step on the relevant data to obtain a result; and
storing the result as [[the]] output to be returned for subsequent invocations of the step taking input identical to the current input, in which case execution of the step is avoided and the stored output is returned for the step.

9. (Original) The method of claim 1, wherein the current input comprises one or more result fields and input parameters.

10. (Currently Amended) A computer-implemented method ~~[[for]]~~ of managing execution of a workflow that is repeatedly executed on data of a database, wherein the workflow is defined by a plurality of steps, each step being an executable function that operates on input from a previous step and produces output for a subsequent step and each step having a defined input format and output format, the method comprising:

receiving current input for execution of a step of the workflow on relevant data of the database, ~~wherein;~~

identifying the step as deterministic, whereby the step generates identical output for given input in repeated executions of the step on the relevant data; and

upon determining that the step has been previously executed using input identical to the current input:

~~without executing the step using the current input, returning output obtained in [[a]] the previous execution of the step using input identical to the current input~~ without executing the step using the received current input; and
upon determining that the step has not been previously executed using input identical to the current input:

executing the step for the current input on the relevant data to obtain a result; and

storing the result to enable managing a next invocation of the step in which the step is passed input identical to the current input, in which case the stored result is returned as output for the step without re-executing the step.

11. (Currently Amended) The method of claim 10, further comprising, ~~prior to returning the output~~ upon determining that the step has been previously executed using input identical to the current input and prior to returning the output:

~~determining whether the step has been previously executed using the input identical to the current input;~~

~~if so,~~ determining whether the relevant data has been changed since the previous execution of the step using the input identical to the current input; and

if the relevant data has not been changed, retrieving the output obtained in the previous execution of the step using the input identical to the current input.

12. (Original) The method of claim 11, wherein determining whether the relevant data has been changed comprises:

determining a timestamp indicating a point of time of the previous execution; and
determining, from a transaction log of the database, whether transactions relative to the relevant data have occurred since the point of time indicated by the timestamp.

13. (Canceled) The method of claim 11, further comprising:

if the step has not been executed using the input identical to the current input:
executing the step for the current input on the relevant data to obtain a result; and

storing the result to enable managing a next invocation of the step in which the step is passed input identical to the current input, in which case the stored result is returned as output for the step without re-executing the step.

14. (Canceled) The method of claim 11, further comprising:

if the relevant data has been changed since the previous execution of the step using the input identical to the current input:

executing the step for the current input on the relevant data to obtain a result; and

storing the result to enable managing a next invocation of the step in which the step is passed input identical to the current input, in which case the stored result is returned as output for the step without re-executing the step.

15. (Currently Amended) A computer readable storage medium containing a program which, when executed by a processor, performs an operation of managing execution of a multi-step workflow that is repeatedly executed on data of a database, wherein the workflow is defined by a plurality of steps, each step being an executable

function that operates on input from a previous step and produces output for a subsequent step and each step having a defined input format and output format, the operation comprising:

receiving current input for execution of a step of the workflow on relevant data of the database, wherein the step has been previously executed on the relevant data using previous input identical to the current input and wherein the previous execution of the step produced previous output;

determining whether the step is deterministic, whereby the step generates identical output for given input in repeated executions of the step on the relevant data; and

if the step is deterministic, returning the previous output produced during the previous execution of the step without re-executing the step.

16. (Currently Amended) The computer readable storage medium of claim 15, wherein the operation further comprises:

inputting the returned previous output to a next sequential step in the workflow.

17. (Currently Amended) The computer readable storage medium of claim 15, wherein determining whether the step is deterministic comprises determining whether a workflow description of the step includes a deterministic flag indicating that the step generates identical output for given input in repeated executions of the step on the relevant data.

18. (Currently Amended) The computer readable storage medium of claim 15, wherein the operation further comprises:

determining whether the current input and the previous input are the same; and
returning the previous output produced during the previous execution of the step only if the current input and the previous input are determined to be the same.

19. (Currently Amended) The computer readable storage medium of claim 18, wherein determining whether the current input and the previous input are the same comprises accessing a hash table representative of the previous input.

20. (Currently Amended) The computer readable storage medium of claim 15, wherein the operation further comprises:

determining whether the relevant data has been changed since the previous execution; and

returning the previous output produced during the previous execution of the step only if the relevant data has not been changed.

21. (Currently Amended) The computer readable storage medium of claim 20, wherein determining whether the relevant data has been changed comprises:

retrieving a timestamp indicating a point of time of the previous execution; and

retrieving a transaction log of the database; and

determining, from the transaction log, whether transactions relative to the relevant data have occurred since the point of time indicated by the timestamp.

22. (Currently Amended) The computer readable storage medium of claim 20, wherein the operation further comprises:

if the relevant data has been changed since the previous execution:

executing the step on the relevant data to obtain a result; and

storing the result as the output to be returned for subsequent invocations of the step taking input identical to the current input, in which case execution of the step is avoided and the stored output is returned for the step.

23. (Currently Amended) The computer readable storage medium of claim 15, wherein the current input comprises one or more result fields and input parameters.

24. (Currently Amended) A computer readable storage medium containing a program which, when executed by a processor, performs an operation of managing execution of a workflow that is repeatedly executed on data of a database, wherein the workflow is defined by a plurality of steps, each step being an executable function that operates on input from a previous step and produces output for a subsequent step and each step having a defined input format and output format, the operation comprising:

receiving current input for execution of a step of the workflow on relevant data of the database, wherein the step generates identical output for given input in repeated executions of the step on the relevant data; and

without executing the step using the current input, returning output obtained in a previous execution of the step using input identical to the current input.

25. (Currently Amended) The computer readable storage medium of claim 24, wherein the operation further comprises, prior to returning the output:

determining whether the step has been previously executed using the input identical to the current input;

if so, determining whether the relevant data has been changed since the previous execution of the step using the input identical to the current input; and

if the relevant data has not been changed, retrieving the output obtained in the previous execution of the step using the input identical to the current input.

26. (Currently Amended) The computer readable storage medium of claim 25, wherein determining whether the relevant data has been changed comprises:

retrieving a timestamp indicating a point of time of the previous execution;

retrieving a transaction log of the database; and

determining, from the transaction log, whether transactions relative to the relevant data have occurred since the point of time indicated by the timestamp.

27. (Currently Amended) The computer readable storage medium of claim 25, wherein the operation further comprises:

if the step has not been executed using the input identical to the current input:

executing the step for the current input on the relevant data to obtain a result; and

storing the result to enable managing a next invocation of the step in which the step is passed input identical to the current input, in which case the stored result is returned as output for the step without re-executing the step.

28. (Currently Amended) The computer readable storage medium of claim 25, wherein the operation further comprises:

if the relevant data has been changed since the previous execution of the step using the input identical to the current input:

executing the step for the current input on the relevant data to obtain a result; and

storing the result to enable managing a next invocation of the step in which the step is passed input identical to the current input, in which case the stored result is returned as output for the step without re-executing the step.

29. (Currently Amended) A computer system, comprising:

a database having data; and

a workflow execution manager residing in memory for managing execution of a multi-step workflow that is repeatedly executed on the data of the database, wherein the workflow is defined by a plurality of steps, each step being an executable function that operates on input from a previous step and produces output for a subsequent step and each step having a defined input format and output format, the workflow execution manager being configured for:

receiving current input for execution of a step of the workflow on relevant data of the database, wherein the step has been previously executed on the relevant data using previous input identical to the current input and wherein the previous execution of the step produced previous output;

determining whether the step is deterministic, whereby the step generates identical output for given input in repeated executions of the step on the relevant data; and

if the step is deterministic, returning the previous output produced during the previous execution of the step without re-executing the step.

30. (Currently Amended) A computer system, comprising:
a database having data; and
a workflow execution manager residing in memory for managing execution of a workflow that is repeatedly executed on the data of the database, wherein the workflow is defined by a plurality of steps, each step being an executable function that operates on input from a previous step and produces output for a subsequent step and each step having a defined input format and output format, the workflow execution manager being configured for:

receiving current input for execution of a step of the workflow on relevant data of the database, wherein the step generates identical output for given input in repeated executions of the step on the relevant data; and

without executing the step using the current input, returning output obtained in a previous execution of the step using input identical to the current input.

31. (Canceled) A data structure residing in memory, comprising:
a specification of at least one step of a multi-step workflow, the specification including:

an indication of at least one functional module adapted for execution of the at least one step; and

a deterministic flag indicating whether the at least one step generates identical output for given input in repeated executions of the step on relevant data of a database.

32. (Currently Amended) A computer-implemented method [[for]] of automatically executing a plurality of functional modules from within an application, comprising:

providing an interface for specifying a single multi-analysis functional module used to execute the plurality of functional modules, whereby user selection of the single multi-analysis functional module is an implicit selection of the plurality of functional modules, and wherein each of the plurality of functional modules is an executable function that operates on input from a previous functional module and produces output for a subsequent functional module and each functional module having a defined input format and output format;

receiving current input for execution of at least one of the functional modules, wherein the at least one functional module has been previously executed using previous input identical to the current input;

determining whether the at least one functional module is deterministic, whereby the at least one functional module generates identical output for given input in repeated executions of the at least one functional module; and

if the at least one functional module is deterministic, returning previous output produced during the previous execution without re-executing the at least one functional module.

33. (Original) The method of claim 32, further comprising retrieving information regarding execution of the plurality of functional modules from a configuration file.

34. (Original) The method of claim 33, wherein determining whether the at least one functional module is deterministic comprises examining information regarding the at least one functional module retrieved from the configuration file.

Please add the following new claim:

35. (New) A computer-implemented method of managing execution of a workflow that is repeatedly executed on data of a database, wherein the workflow is defined by a plurality of steps, each step being an executable function that operates on input from a previous step and produces output for a subsequent step and each step having a defined input format and output format, the method comprising:

receiving current input for execution of a step of the workflow on relevant data of the database;

identifying the step as deterministic, whereby the step generates identical output for given input in repeated executions of the step on the relevant data;

upon determining that the step has been previously executed using input identical to the current input, determining whether the relevant data has been changed since the previous execution of the step using the input identical to the current input;

if the relevant data has been changed since the previous execution of the step using the input identical to the current input:

executing the step for the current input on the relevant data to obtain a result; and

storing the result to enable managing a next invocation of the step in which the step is passed input identical to the current input, in which case the stored result is returned as output for the step without re-executing the step; and
if the relevant data has not been changed since the previous execution of the step using the input identical to the current input:

returning output obtained in the previous execution of the step using the input identical to the current input without executing the step using the received current input.